WHAT IS CLAIMED IS:

1. A parallel encryption method for providing both data confidentiality and integrity for a message, comprising the steps of:

receiving an input plaintext string comprising a message;

generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks;

presenting the equal-size blocks and the MDC block to a selected parallel encryption mode that makes one and only one processing pass with a single cryptographic primitive over each of the said equal-size blocks and said MDC block to create a plurality of hidden ciphertext blocks each of ℓ bits in length; and

performing a hidden ciphertext randomization function over said plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length.

2. The method as defined in claim 1, comprising the steps of:

wherein said selected parallel encryption mode is confidentiality-secure against chosen-plaintext attacks, wherein each of said equal-size blocks and the MDC block is processed by a block cipher using a secret key (K) to obtain said plurality of hidden ciphertext blocks; and

wherein said performing a hidden ciphertext randomization function step comprises combining each of said hidden ciphertext blocks with a corresponding element of a sequence of unpredictable elements for the hidden ciphertext to create a set of output blocks of the ciphertext, wherein a hidden ciphertext block identified by an index i is combined

with the element of the sequence identified by index i by an operation for the hidden ciphertext that has an inverse.

3. The method as defined in claim 2, wherein said selected parallel encryption mode that is confidentiality-secure against chosen-plaintext attacks comprises the steps of:

performing a plaintext randomization function over said plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; and

processing each of said hidden plaintext blocks by a block cipher using said secret key (K) to obtain said plurality of hidden ciphertext blocks.

4. The method as defined in claim 3, wherein said performing a plaintext randomization function step comprises combining each of said equal-size blocks and the MDC block with a corresponding element of a sequence of unpredictable elements for the hidden plaintext to create a set of hidden plaintext blocks, wherein an equal-size block or the MDC block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden plaintext that has an inverse.

5. The method as defined in claim 2,

wherein the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by the inverse operation of the operation for the hidden ciphertext is unpredictable; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden ciphertext are any two different elements of the same sequence of unpredictable elements for the hidden ciphertext used for the encryption of said plaintext string; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden ciphertext are any two different elements of a plurality of sequences of unpredictable elements for the hidden ciphertext used for encryption of a plurality of plaintext strings with the same secret key K.

6. The method as defined in claim 4,

wherein the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by the inverse operation of the operation for the hidden plaintext is unpredictable; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden plaintext are any two different elements of the same sequence of unpredictable elements for the hidden plaintext used for the encryption of said plaintext string; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden plaintext are any two different elements of a plurality of sequences of unpredictable elements for the hidden plaintext used for encryption of a plurality of plaintext strings with the same secret key K.

7. The method as defined in claim 4,

wherein any two different unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext are not pair-wise independent;

wherein said unpredictable elements selected as said two unpredictable elements for the hidden ciphertext are any two different elements of the same sequence of unpredictable elements for the hidden ciphertext used for the encryption of said plaintext string; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden ciphertext are any two different elements of a plurality of sequences of unpredictable elements for the hidden ciphertext used for encryption of a plurality of plaintext strings with the same secret key K; and

wherein any two different unpredictable elements of the sequence of unpredictable elements for the hidden plaintext are not pair-wise independent;

wherein said unpredictable elements selected as said two unpredictable elements for the hidden plaintext are any two different elements of the same sequence of unpredictable elements for the hidden plaintext used for the encryption of said plaintext string; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden plaintext are any two different elements of a plurality of sequences of unpredictable elements for the hidden plaintext used for encryption of a plurality of plaintext strings with the same secret key K.

8. The method as defined in claim 1, wherein said creating an MDC block step comprises applying the non-cryptographic MDC function to the equal-sized blocks of the plaintext.

9. The method of claim 8, wherein said non-cryptographic MDC function is the bit-wise exclusive-or function.

10. The method of claim 8, wherein said non-cryptographic MDC function is the addition modulo 2^t - 1function.

11. The method of claim 8, wherein said non-cryptographic MDC function is the subtraction modulo 2^t - 1function.

12. The method of claim 8 further comprising combining the result from applying the non-cryptographic Manipulation Detection Code function to the plurality of equal-sized blocks of the plaintext with a secret, ℓ-bit random vector generated on a per-message basis to obtain said MDC block.

13. The method as defined in claim 12, wherein said combining step comprises performing the combination using a bit-wise exclusive-or function.

14. The method as defined in claim 12, wherein said combining step comprises performing the combination using addition modulo $2^{\ell} - 1$.

15. The method as defined in claim 12, wherein said combining step comprises performing the combination using subtraction modulo $2^{\ell} - 1$.

16. The method as defined in claim 12, comprising the step of generating said secret random vector from a secret random number generated on a per-message basis.

17. The method as defined in claim 2, further comprising the step of appending the created MDC block after a last block of the set of equal-sized blocks of the plaintext.

18. The method as defined in claim 2, wherein the hidden ciphertext blocks from the processing step comprise n + 1 hidden ciphertext blocks each of ℓ-bit length, where n is the total number of blocks in said set of equal-sized blocks of the plaintext.

19. The method as defined in claim 2, further comprising the step of generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden ciphertext by combining a different element identifier for each of the unpredictable elements and a secret random number.

20. The method as defined in claim 4, further comprising the step of generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden plaintext by combining a different element identifier for each of the unpredictable elements and a secret random number.

21. The method as defined in claim 4, further comprising the steps of:

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden ciphertext by combining a different element identifier for each of the unpredictable elements and a secret random number; and

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden plaintext by combining a different element identifier for each of the unpredictable elements and said secret random number.

22. The method as defined in claim 21, further comprising the steps of:

wherein generating each element in said sequence of unpredictable elements for the hidden ciphertext comprises a modular 2^{ℓ} multiplication of a different unique element identifier (i) for each element in the sequence of unpredictable elements and said secret random number; and

wherein generating each element in said sequence of unpredictable elements for the hidden plaintext comprises a modular 2^{ℓ} multiplication of a different unique element identifier (i) for each element in the sequence of unpredictable elements and said secret random number for all the equal-size blocks of the plaintext and by modular 2^{ℓ} multiplication of n + 2 and said secret random number for the MDC block.

23. The method as defined in claim 21, further comprising the steps of:

enciphering the secret random number using the block cipher using the secret key (K); and

including this enciphered secret random number (y_0) as one of said output ciphertext blocks.

24. The method of claim 21, wherein the secret random number is provided by a random number generator.

25. The method as defined in claim 21, further comprising:

generating said secret random number by enciphering a count of a counter initialized to a constant, said enciphering being performed with the block cipher using the secret key (K); and

incrementing said counter by one on every message encryption.

26. The method as defined in claim 25, wherein said counter is initialized to a constant whose value is the ℓ-bit representation of negative one.

27. The method as defined in claim 25, comprising:

initializing said counter to a secret value of ℓ bits in length.

28. The method as defined in claim 25, further comprising:

outputting said counter value as an output block of the encryption mode.

29. The method as defined in claim 4, further comprising the steps of:

deriving a block-index-independent unpredictable element;

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden ciphertext by combining said block-index-independent unpredictable element with each of a plurality of block-index-dependent unpredictable elements for the hidden ciphertext; and

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden plaintext by combining said block-index-independent unpredictable element with each of a plurality of block-index-dependent unpredictable elements for the hidden ciphertext.

30. The method of claim 29, further comprising the steps of:

wherein said block-index-independent unpredictable element is obtained from a count of an ℓ -bit counter initialized to a non-zero constant, and a per-key secret, first random initial number shared between sender and receiver; and

wherein each of said plurality of block-index-dependent unpredictable elements for the hidden ciphertext is obtained from an ℓ -bit element index and a secret, second random initial number shared between sender and receiver;

wherein each of said plurality of block-index-dependent unpredictable elements for the hidden plaintext is obtained from an ℓ -bit

element index and a per-key secret, second random initial number shared between sender and receiver;

wherein said secret, first and second random initial numbers are independent; and

wherein said ℓ-bit counter is incremented by one on every message encryption.

31. The method of claim 29, wherein said combining to obtain the unpredictable elements for the hidden ciphertext comprises an addition modulo 2^{ℓ}.

32. The method of claim 29, wherein said combining to obtain the unpredictable elements for the hidden plaintext comprises an addition modulo 2^{ℓ}.

33. The method of claim 29, wherein said combining to obtain the unpredictable elements for the hidden ciphertext comprises a subtraction modulo 2^{ℓ}.

34. The method of claim 29, wherein said combining to obtain the unpredictable elements for the hidden plaintext comprises a subtraction modulo 2^{ℓ}.

35. The method of claim 29, wherein said combining to obtain the unpredictable elements for the hidden ciphertext comprises a bit-wise exclusive-or operation.

36. The method of claim 29, wherein said combining to obtain the unpredictable elements for the hidden plaintext comprises a bit-wise exclusive-or operation.

37. The method of claim 30, further comprising the steps of:

wherein said block-index-independent unpredictable element is obtained by multiplication modulo 2^t of said secret, first random initial number with a different value of the counter; and

wherein each of said plurality of block-index-dependent unpredictable elements for the hidden ciphertext is obtained by multiplication modulo 2^t of said secret, second random initial number with the index i of the hidden ciphertext block; and

wherein each of said plurality of block-index-dependent unpredictable elements for the hidden plaintext is obtained by multiplication modulo 2^t of said secret, second random initial number with the index i of the plaintext block; and

wherein the unpredictable element for the hidden plaintext corresponding to the MDC block is the block-index-independent unpredictable element itself.

38. The method as defined in claim 2, wherein said operation for the hidden ciphertext that has an inverse is the addition modulo 2^t.

39. The method as defined in claim 2, wherein said operation for the hidden ciphertext that has an inverse is a bit-wise exclusive-or operation.

40. The method as defined in claim 2, wherein said operation for the hidden ciphertext that has an inverse is the subtraction modulo 2^{ℓ} operation.

41. The method as defined in claim 4, wherein said operation for the hidden plaintext that has an inverse is the addition modulo 2^{ℓ}.

42. The method as defined in claim 4, wherein said operation for the hidden plaintext that has an inverse is a bit-wise exclusive-or operation.

43. The method as defined in claim 4, wherein said operation for the hidden plaintext that has an inverse is the subtraction modulo 2^{ℓ} operation.

44. The method as defined in claim 1, wherein said generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises the steps of:

padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and

partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

45. The method as defined in claim 44, wherein said padding of the input plaintext string is a standard padding method.

46. The method as defined in claim 44, wherein said padding of the input plaintext string comprises the steps of:

if the last block of the plaintext has ℓ bits in length derive a last element of said sequence of unpredictable elements for the hidden

plaintext to be combined with the MDC block to form a hidden plaintext block from the bit-wise complement of a random number;

else, append to the last block of the plaintext the bit 1 and the necessary bits of 0 to generate a last equal-size block, and derive a last element of said sequence of unpredictable elements for the hidden plaintext to be combined with the MDC block to form a hidden plaintext block from said random number; and

generating each but the last of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden plaintext by combining a different element identifier for each of the unpredictable elements and said secret random number.

47. The method as defined in claim 44, wherein said padding of the input plaintext string comprises the steps of:

if the last block of the plaintext has ℓ bits in length derive a last element of said sequence of unpredictable elements for the hidden plaintext to be combined with the MDC block to form a hidden plaintext block from a different block-index-independent unpredictable element obtained from the bit-wise complement of a first random number shared between a sender and a receiver;

else, append to the last block of the plaintext the bit 1 and the necessary bits of 0 to generate a last equal-size block, and derive the last element of said sequence of unpredictable elements for the hidden plaintext to be combined with the MDC block to form a hidden plaintext block from a different block-index-independent unpredictable element obtained from the said first random number shared between a sender and a receiver; and

generating each but the last of a plurality of the unpredictable elements of said sequence of unpredictable elements for the

hidden plaintext by combining a different block-index-independent unpredictable element obtained from said first random number shared between a sender and a receiver and each of a plurality of block-index-dependent unpredictable elements for the hidden plaintext.

48. A parallel decryption method that is the inverse of the parallel encryption method which provides both data confidentiality and integrity, comprising the steps of:

presenting a string including ciphertext string for decryption;

partitioning said ciphertext string into a plurality of ciphertext blocks comprising ℓ bits each;

selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length;

presenting the hidden ciphertext blocks to a selected parallel decryption mode that makes one and only one processing pass with a single cryptographic primitive that is the inverse of an encryption single cryptographic primitive over the plurality of hidden ciphertext blocks to obtain a plurality of plaintext blocks and one decrypted MDC block each of ℓ bits in length;

verifying integrity of the plaintext blocks using a non-cryptographic Manipulation Detection Function (MDC) function;

outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and

outputting a failure indicator if the integrity verification fails.

49. The method as defined in claim 48, wherein performing said reverse hidden-ciphertext randomization function comprises:

generating a sequence of unpredictable elements for the hidden ciphertext each of ℓ-bit length in the same manner as used at an encryption method;

selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block in the same order as that used at an encryption method, and combining said selected ciphertext blocks with said sequence of unpredictable elements for the hidden ciphertext to obtain a plurality of hidden ciphertext blocks (z_i), such that each of the $n+1$ ciphertext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden ciphertext identified by index i, by the inverse of said operation for the hidden ciphertext used at the encryption method; and

wherein the verifying integrity step comprises creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks; and comparing said created MDC decryption block with the decrypted MDC block.

50. The method as defined in claim 49, wherein said creating an MDC decryption block further comprises combining the result with a secret, ℓ-bit random vector, said combining operation being the same as the combining operation at the encryption method, and said secret random vector being derived from said secret random number in the same manner as at the encryption method.

51. The method as defined in claim 48, wherein said selected parallel decryption mode comprises the steps of:

processing each of said hidden ciphertext blocks with the inverse of the block cipher used at an encryption method using a secret key (K) to obtain a plurality of hidden plaintext blocks; and

performing a reverse plaintext randomization function over said plurality of hidden plaintext blocks to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ bits in length.

52. The method as defined in claim 51, wherein performing said reverse plaintext randomization function comprises:

generating a sequence of unpredictable elements for the hidden plaintext each of ℓ-bit length in the same manner as used at an encryption method; and

combining said selected hidden plaintext blocks with said sequence of unpredictable elements for the hidden plaintext to obtain a plurality of n plaintext blocks and one decrypted MDC block, such that each of the $n+1$ hidden plaintext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden plaintext identified by index i, by the inverse of said operation for the hidden plaintext used at the encryption method.

53. The method of claim 49, further comprising the steps of:

deriving a secret random number from said ciphertext string presented for decryption; and

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden ciphertext in the same manner as at the encryption method.

54. The method of claim 52, further comprising the steps of:

deriving a secret random number from said ciphertext string presented for decryption; and

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden plaintext in the same manner as at the encryption method.

55. The method of claim 52, further comprising the steps of:

deriving a secret random number from said ciphertext string presented for decryption;

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden ciphertext in the same manner as at the encryption method; and

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden plaintext in the same manner as at the encryption method.

56. The method of claim 48, further comprising:

selecting the ciphertext block of a secret random number (y_0) from said string presented for decryption; and

deciphering the selected ciphertext block to obtain the secret random number.

57. The method as defined in claim 56, wherein said deciphering step comprises performing the deciphering with the inverse of the said block cipher using the secret key (K).

58. The method of claim 48, further comprising:

for the encryption method generating a secret random number by enciphering a count of a counter initialized to a constant, said enciphering being performed with the block cipher using the secret key; and

incrementing said counter by one on every message encryption; and

further comprising for decrypting the ciphertext blocks of the partitioned ciphertext string the steps of:

selecting a counter block representing the count of the counter from said string presented at decryption; and

enciphering said selected counter block to obtain the secret random number.

59. The method as defined in claim 58, wherein the enciphering step comprises performing said enciphering with the block cipher using the secret key.

60. The method of claim 48, further comprising:

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden ciphertext by combining a different block-index-independent unpredictable element with each of a plurality of block-index-dependent unpredictable elements for the hidden ciphertext in the same manner as at the encryption method; and

generating each of a plurality of the unpredictable elements of said sequence of unpredictable elements for the hidden plaintext by combining a different block-index-independent unpredictable element with each of a plurality of block-index-dependent unpredictable elements for the hidden plaintext in the same manner as at the encryption method.

61. The method as defined in claim 48, wherein the string presented for decryption is obtained by applying the encryption method that provides both data confidentiality and integrity to an input plaintext string, further comprising:

outputting said input plaintext string.

62. A method for segmented encryption processing of a message comprising the steps of:

partitioning said input plaintext string into a plurality of input plaintext segments;

concurrently presenting each different one of said plurality of input plaintext segments to a different one of a plurality of parallel encryption methods, each of said different methods using a different ℓ-bit secret random number per segment to obtain a ciphertext segment, wherein each encryption method provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, and uses a non-cryptographic Manipulation Detection Code function, wherein said single cryptographic primitive is an ℓ-bit block cipher using a secret key;

assembling the plurality of ciphertext segments into a ciphertext string; and

outputting the ciphertext string.

63. The method as defined in claim 62, wherein said assembling step comprises including in the ciphertext string the number of ciphertext segments, a ciphertext segment index, a length of each ciphertext segment and a sequence of ciphertext segments.

64. The method of claim 62, further comprising:

generating said different ℓ-bit secret random number per segment from a secret random number of ℓ bits in length.

65. The method of claim 64, further comprising:

generating said different secret random number per segment from the secret random number of ℓ bits by adding modulo 2^{ℓ} a plaintext segment sequence index for that segment to the secret random number.

66. The method of claim 64, further comprising:

generating said secret random number of ℓ bits in length by a random number generator;

enciphering said secret random number with said block cipher using a first key (K); and

including the enciphered secret random number as an output block of said output ciphertext string.

67. The method of claim 62, further comprising:

generating each of the said secret random number per segment by enciphering the result of adding the segment number to a counter initialized to a constant, said enciphering being done with said block cipher using said first key (K); and

outputting said counter value as an output block of said output ciphertext string; and

incrementing after every different message encryption said counter by a number equal to a number of plaintext segments in the message.

68. The method of claim 62, further comprising:

generating each of the said secret random number per segment from a per-key secret, first random initial number shared between sender and receiver and the result of adding modulo 2^{ℓ} the segment number to a counter initialized to a constant; and

outputting said counter value as an output block of said output ciphertext string; and

incrementing after every different message encryption said counter by a number equal to a number of plaintext segments in the message.

69. The method of claim 68, wherein said generating each of the said secret random number per segment comprises multiplying modulo 2^{ℓ} said per-key secret, first random initial number shared between sender and receiver with the result of adding the segment number to said counter.

70. A method for segmented decryption processing of a message comprising the steps of:

presenting a string including the ciphertext string of a message for decryption;

partitioning said ciphertext string into a plurality of ciphertext segments;

concurrently presenting said plurality of ciphertext segments to a plurality of decryption modes;

obtaining a different secret random number per ciphertext segment in the same manner as at the segmented encryption method;

decrypting each ciphertext segment using said different secret random number per ciphertext segment to obtain a plaintext segment, using a parallel decryption method that is the inverse of the parallel encryption method that provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, wherein said single cryptographic primitive is an ℓ-bit block cipher using a secret key, and using a non-cryptographic Manipulation Detection Code function for verifying integrity of the plaintext blocks of each plaintext segment; and

verifying the integrity of each plaintext segment and for each plaintext segment, outputting either the plaintext segment if the integrity verification passes, or an error indicator.

71. The method of claim 70, wherein each of the said different secret random numbers per ciphertext segment are obtained from a secret random number in the same manner at as used at a segmented encryption method.

72. The method of claim 71, further comprising:

selecting a ciphertext block of the secret random number from said string presented for decryption;

deciphering the selected ciphertext block to obtain the secret random number.

73. The method as defined in claim 72, performing said deciphering step with the inverse of a block cipher using said secret key, said block cipher and said secret key being the same as to those used at a segmented encryption method.

74. The method of claim 70, further comprising:

for the segmented encryption method generating said secret random number per ciphertext segment by enciphering the result of adding modulo 2^{ℓ} the segment number with a counter initialized to a constant, said enciphering being done with said block cipher using said first key (K); and

incrementing after every different message encryption said counter by a number equal to a number of plaintext segments in the message; and

further comprising for segmented decryption of the ciphertext segments of the partitioned ciphertext string the steps of:

selecting a counter block holding the count of the counter from said string presented for decryption;

enciphering the result of adding modulo 2^{ℓ} the segment number with said selected counter block to obtain said secret random number per ciphertext segment.

75. The method as defined in claim 74, wherein said enciphering of the result of adding modulo 2^{ℓ} the segment number with a counter initialized to a constant step comprises enciphering with the block cipher using the same key as that used for segmented encryption.

76. The method of claim 70, further comprising:

for the segmented encryption method generating each of the said secret random number per segment from a per-key secret, first random initial number shared between sender and receiver and the result of adding modulo 2^{ℓ} the segment number to a counter initialized to a constant; and

outputting said counter value as an output block of said output ciphertext string; and

incrementing after every different message encryption said counter by a number equal to a number of plaintext segments in the message; and

further comprising for segmented decryption of the ciphertext segments of the partitioned ciphertext string the steps of:

selecting a counter block holding the count of the counter from said string presented for decryption; and

generating each of the said secret random number per ciphertext segment from said per-key secret, first random initial number shared between sender and receiver and the result of adding modulo 2^{ℓ} the segment number to said counter.

77. A parallel encryption method for providing both data confidentiality and integrity for a message, that updates a ciphertext string incrementally, comprising the steps of:

receiving an input plaintext string comprising a message;

generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks;

performing a plaintext randomization function over said plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length;

processing each of said hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks;

performing a hidden ciphertext randomization function over said plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length; and

further comprising the steps of:

receiving an input plaintext string;

generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

receiving an input ciphertext string including a plurality of $n+1$ equal-size blocks of the ciphertext of ℓ bits in length, wherein the $n+1$ block of the ciphertext corresponds to an MDC block for said plaintext string;

receiving a new ℓ-bit plaintext block to replace an ℓ-bit plaintext block at index i;

creating a new MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks and the new ℓ-bit plaintext block;

performing the same plaintext randomization function as that used at a parallel encryption method over said new ℓ-bit plaintext block and the new MDC block to create two new hidden plaintext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index $n+1$ for the new MDC block;

processing each of said two new hidden plaintext blocks by a block cipher using said secret key (K) to obtain two new hidden ciphertext blocks;

performing the same hidden ciphertext randomization function as that used at a parallel encryption method over said two new hidden ciphertext blocks to create two new output ciphertext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index $n+1$ for the new MDC block;

replacing in the input ciphertext string, the input ciphertext block at index i with the output ciphertext block for the new ℓ-bit plaintext block and replace the input ciphertext block at index $n+1$ with the output ciphertext block for the new MDC block, to create a new ciphertext string; and

outputting the new ciphertext string.

78. The method as defined in claim 77, wherein said generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises the steps of:

padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and

partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

79. The method of claim 77 comprising the steps of:

receiving a plurality of new ℓ-bit plaintext blocks to replace a plurality of ℓ-bit plaintext blocks at said plaintext string at index i; and

providing a parallel encryption method that outputs a ciphertext string incrementally for each of the said plurality of new ℓ-bit plaintext blocks.

80. A parallel encryption method for providing both data confidentiality and integrity for a message, comprising the steps of:

receiving an input plaintext string comprising a message;

generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length;

creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks;

performing a plaintext randomization function over said plurality of equal-sized blocks of the plaintext and the MDC block using a different plaintext index for each equal-sized block and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length;

processing each of said hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks;

performing a hidden ciphertext randomization function over said plurality of hidden ciphertext blocks using a different ciphertext index

for each hidden ciphertext block to create a plurality of output ciphertext blocks each of ℓ bits in length; and

further providing an out-of-order decryption method for the parallel encryption method, which provides both data confidentiality and integrity, comprising the steps of:

receiving a string including a plurality of $n+1$ ℓ-bit ciphertext blocks for decryption;

selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks using said ciphertext index to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length;

processing each of said hidden ciphertext blocks with the inverse of the block cipher used at an encryption method using said secret key (K) to obtain a plurality of hidden plaintext blocks; and

performing an inverse plaintext randomization function over said plurality of hidden plaintext blocks using said plaintext index to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ-bit length;

creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks in the same manner as at a parallel encryption method;

verifying integrity of the plaintext blocks by comparing said created MDC decryption block with the decrypted MDC block;

outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and

outputting a failure indicator if the integrity verification fails.

81. The method as defined in claim 80, wherein said generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises the steps of:

padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and

partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

82. A program product for parallel encryption for providing both data confidentiality and integrity for a message, including machine-readable program code for causing a machine to perform the following method steps:

receiving an input plaintext string comprising a message;

generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks;

presenting the equal-size blocks and the MDC block to a selected parallel encryption mode that makes one and only one processing pass with a single cryptographic primitive over each of the said equal-size blocks and said MDC block to create a plurality of hidden ciphertext blocks each of ℓ bits in length; and

performing a hidden ciphertext randomization function over said plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length.

83. The program product as defined in claim 82, wherein the program code includes code

to cause the step of presenting the equal-size blocks and the MDC block to a selected parallel encryption mode processing each of said equal-size blocks and the MDC block by a parallel encryption mode to be confidentiality-secure against chosen-plaintext attacks, wherein each of said equal-size blocks and the MDC block is processed by a block cipher using a secret key (K) to obtain said plurality of hidden ciphertext blocks; and

to cause the step of performing a hidden ciphertext randomization function step comprises code for combining each of said hidden ciphertext blocks with a corresponding element of a sequence of unpredictable elements for the hidden ciphertext to create a set of output blocks of the ciphertext, wherein a hidden ciphertext block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden ciphertext that has an inverse.

84. The program product as defined in claim 83, wherein the program code for causing the performance of the step of processing each of said the equal-size blocks and the MDC block by a parallel encryption mode that is confidentiality-secure against chosen-plaintext attacks comprises code for:

performing a plaintext randomization function over said plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; and

processing each of said hidden plaintext blocks by a block cipher using said secret key (K) to obtain said plurality of hidden ciphertext blocks.

85. The program product as defined in claim 84, wherein the program code for performing a plaintext randomization function step comprises code for combining each of said equal-size blocks and the MDC

block with a corresponding element of a sequence of unpredictable elements for the hidden plaintext to create a set of hidden plaintext blocks, wherein an equal-size block or the MDC block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden plaintext that has an inverse.

86. The program product as defined in claim 83,

wherein the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by the inverse operation of the operation for the hidden ciphertext is unpredictable; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden ciphertext are any two different elements of the same sequence of unpredictable elements for the hidden ciphertext used for the encryption of said plaintext string; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden ciphertext are any two different elements of a plurality of sequences of unpredictable elements for the hidden ciphertext used for encryption of a plurality of plaintext strings with the same secret key K.

87. The program product as defined in claim 85,

wherein the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by the inverse operation of the operation for the hidden plaintext is unpredictable; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden plaintext are any two different elements of the same sequence of unpredictable elements for the hidden plaintext used for the encryption of said plaintext string; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden plaintext are any two different elements of a plurality of sequences of unpredictable elements for the hidden plaintext used for encryption of a plurality of plaintext strings with the same secret key K.

88. A program product for parallel decryption that is the inverse of a program product for parallel encryption which provides both data confidentiality and integrity, comprising machine-readable program code for causing a machine to perform the following method steps:

presenting a string including ciphertext string for decryption;

partitioning said ciphertext string into a plurality of ciphertext blocks comprising ℓ bits each;

selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length;

presenting the hidden ciphertext blocks to a selected parallel decryption mode that makes one and only one processing pass with a single cryptographic primitive that is the inverse of an encryption single cryptographic primitive over the plurality of hidden ciphertext blocks to obtain a plurality of plaintext blocks and one decrypted MDC block each of ℓ bits in length;

verifying integrity of the plaintext blocks using a non-cryptographic Manipulation Detection Function (MDC) function;

outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and

outputting a failure indicator if the integrity verification fails.

89. The program product as defined in claim 88, wherein said program code for causing the performance of the step of selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block and performing said reverse hidden-ciphertext randomization function comprises code for:

generating a sequence of unpredictable elements for the hidden ciphertext each of ℓ-bit length in the same manner as used at an encryption program product;

selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block in the same order as that used at an encryption program product, and combining said selected ciphertext blocks with said sequence of unpredictable elements for the hidden ciphertext to obtain a plurality of hidden ciphertext blocks (z_i), such that each of the $n+1$ ciphertext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden ciphertext identified by index i, by the inverse of said operation for the hidden ciphertext used at the encryption program product; and

wherein the program code for causing the performance of the step of verifying integrity comprises code for creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks; and code for comparing said created MDC decryption block with the decrypted MDC block.

90. The program product as defined in claim 88, wherein said program code for causing the performance of the step of presenting the hidden ciphertext blocks to a selected parallel decryption mode comprises code for:

processing each of said hidden ciphertext blocks with the inverse of the block cipher used at an encryption program product using a secret key (K) to obtain a plurality of hidden plaintext blocks; and

performing a reverse plaintext randomization function over said plurality of hidden plaintext blocks to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ bits in length.

91. The program product as defined in claim 90, wherein said program code for causing the performance of said reverse plaintext randomization function comprises code for:

generating a sequence of unpredictable elements for the hidden plaintext each of ℓ-bit length in the same manner as used at an encryption program product; and

combining said selected hidden plaintext blocks with said sequence of unpredictable elements for the hidden plaintext to obtain a plurality of n plaintext blocks and one decrypted MDC block, such that each of the $n+1$ hidden plaintext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden plaintext identified by index i, by the inverse of said operation for the hidden plaintext used at the encryption program product.

92. A program product for segmented encryption processing of a message comprising machine-readable program code for causing the performance of the following method steps:

partitioning said input plaintext string into a plurality of input plaintext segments;

concurrently presenting each different one of said plurality of input plaintext segments to a different one of a plurality of program products for parallel encryption, each of said different program products

using a different ℓ-bit secret random number per segment to obtain a ciphertext segment, wherein each encryption program product provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, and uses a non-cryptographic Manipulation Detection Code function, wherein said single cryptographic primitive is an ℓ-bit block cipher using a secret key;

assembling the plurality of ciphertext segments into a ciphertext string; and

outputting the ciphertext string.

93. The program product as defined in claim 92, wherein said program code for causing the performance of the step of assembling comprises code for including in the ciphertext string the number of ciphertext segments, a ciphertext segment index, a length of each ciphertext segment and a sequence of ciphertext segments.

94. A program product for segmented decryption processing of a message comprising machine-readable program code for causing a machine to perform the following method steps:

presenting a string including the ciphertext string of a message for decryption;

partitioning said ciphertext string into a plurality of ciphertext segments;

concurrently presenting said plurality of ciphertext segments to a plurality of decryption modes;

obtaining a different secret random number per ciphertext segment in the same manner as at the program product for segmented encryption;

for decrypting each ciphertext segment using said different secret random number per ciphertext segment to obtain a plaintext

segment, using a program product for parallel decryption that is the inverse of a program product for parallel encryption that provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, wherein said single cryptographic primitive is an ℓ-bit block cipher using a secret key, and using a non-cryptographic Manipulation Detection Code function for verifying integrity of the plaintext blocks of each plaintext segment; and

verifying the integrity of each plaintext segment and for each plaintext segment, outputting either the plaintext segment if the integrity verification passes, or an error indicator.

95. A system for parallel encryption for providing both data confidentiality and integrity for a message, comprising:

a first component for receiving an input plaintext string comprising a message;

a second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

a third component for creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks;

a fourth component for presenting the equal-size blocks and the MDC block to a selected parallel encryption mode that makes one and only one processing pass with a single cryptographic primitive over each of the said equal-size blocks and said MDC block to create a plurality of hidden ciphertext blocks each of ℓ bits in length; and

a fifth component for performing a hidden ciphertext randomization function over said plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length.

96. The system as defined in claim 95, comprising:

wherein said fourth component for presenting the equal-size blocks and the MDC block to a selected parallel encryption mode comprises a component for processing each of said the equal-size blocks and the MDC block by a parallel encryption mode is confidentiality-secure against chosen-plaintext attacks, wherein each of said equal-size blocks and the MDC block is processed by a block cipher using a secret key (K) to obtain said plurality of hidden ciphertext blocks; and

wherein said fifth component for performing a hidden ciphertext randomization function step comprises a component for combining each of said hidden ciphertext blocks with a corresponding element of a sequence of unpredictable elements for the hidden ciphertext to create a set of output blocks of the ciphertext, wherein a hidden ciphertext block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden ciphertext that has an inverse.

97. The system as defined in claim 96, wherein said component for processing each of said the equal-size blocks and the MDC block by a parallel encryption mode that is confidentiality-secure against chosen-plaintext attacks comprises:

a component for performing a plaintext randomization function over said plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; and

a component for processing each of said hidden plaintext blocks by a block cipher using said secret key (K) to obtain said plurality of hidden ciphertext blocks.

98. The system as defined in claim 97, wherein said component for performing a plaintext randomization function step comprises a component for combining each of said equal-size blocks and the MDC block with a corresponding element of a sequence of unpredictable elements for the hidden plaintext to create a set of hidden plaintext blocks, wherein an equal-size block or the MDC block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden plaintext that has an inverse.

99. The system as defined in claim 96,

wherein the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by the inverse operation of the operation for the hidden ciphertext is unpredictable; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden ciphertext are any two different elements of the same sequence of unpredictable elements for the hidden ciphertext used for the encryption of said plaintext string; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden ciphertext are any two different elements of a plurality of sequences of unpredictable elements for the hidden ciphertext used for encryption of a plurality of plaintext strings with the same secret key K.

100. The system as defined in claim 98,

wherein the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by the inverse operation of the operation for the hidden plaintext is unpredictable; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden plaintext are any two different elements of the same sequence of unpredictable elements for the hidden plaintext used for the encryption of said plaintext string; and

wherein said unpredictable elements selected as said two unpredictable elements for the hidden plaintext are any two different elements of a plurality of sequences of unpredictable elements for the hidden plaintext used for encryption of a plurality of plaintext strings with the same secret key K.

101. A system for parallel decryption that is the inverse of a system for parallel encryption which provides both data confidentiality and integrity, comprising:

a first component for presenting a string including ciphertext string for decryption;

a second component for partitioning said ciphertext string into a plurality of ciphertext blocks comprising ℓ bits each;

a third component for selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length;

a fourth component for presenting the hidden ciphertext blocks to a selected parallel decryption mode that makes one and only one processing pass with a single cryptographic primitive that is the inverse of an encryption single cryptographic primitive over the plurality of hidden ciphertext blocks to obtain a plurality of plaintext blocks and one decrypted MDC block each of ℓ bits in length;

a fifth component for verifying integrity of the plaintext blocks using a non-cryptographic Manipulation Detection Function (MDC) function;

a sixth component for outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and

a seventh component for outputting a failure indicator if the integrity verification fails.

102. The system as defined in claim 101, wherein said third component for selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block and performing said reverse hidden-ciphertext randomization function comprises:

a component for generating a sequence of unpredictable elements for the hidden ciphertext each of ℓ-bit length in the same manner as used at an encryption system;

a component for selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block in the same order as that used at an encryption system, and combining said selected ciphertext blocks with said sequence of unpredictable elements for the hidden ciphertext to obtain a plurality of hidden ciphertext blocks (z_i), such that each of the $n+1$ ciphertext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden ciphertext identified by index i, by the inverse of said operation for the hidden ciphertext used at the encryption system; and

wherein the fifth code for verifying integrity step comprises a component for creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted

plaintext data blocks; and a component for comparing said created MDC decryption block with the decrypted MDC block.

103. The system as defined in claim 101, wherein said fourth component for presenting the hidden ciphertext blocks to a selected parallel decryption mode comprises:

a component for processing each of said hidden ciphertext blocks with the inverse of the block cipher used at an encryption system using a secret key (K) to obtain a plurality of hidden plaintext blocks; and

a component for performing a reverse plaintext randomization function over said plurality of hidden plaintext blocks to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block of ℓ bits in length.

104. The system as defined in claim 103, wherein said component for performing said reverse plaintext randomization function comprises:

a component for generating a sequence of unpredictable elements for the hidden plaintext each of ℓ-bit length in the same manner as used at an encryption system; and

a component for combining said selected hidden plaintext blocks with said sequence of unpredictable elements for the hidden plaintext to obtain a plurality of n plaintext blocks and one decrypted MDC block, such that each of the n+1 hidden plaintext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden plaintext identified by index i, by the inverse of said operation for the hidden plaintext used at the encryption system.

105. A system for segmented encryption processing of a message comprising:

a first component for partitioning said input plaintext string into a plurality of input plaintext segments;

a second component for concurrently presenting each different one of said plurality of input plaintext segments to a different one of a plurality of systems for parallel encryption, each of said different systems using a different ℓ-bit secret random number per segment to obtain a ciphertext segment, wherein each encryption system provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, and uses a non-cryptographic Manipulation Detection Code function, wherein said single cryptographic primitive is an ℓ-bit block cipher using a secret key;

a third component for assembling the plurality of ciphertext segments into a ciphertext string; and

a fourth component outputting the ciphertext string.

106. The system as defined in claim 105, wherein said third component for assembling step comprises a component for including in the ciphertext string the number of ciphertext segments, a ciphertext segment index, a length of each ciphertext segment and a sequence of ciphertext segments.

107. A system for segmented decryption processing of a message comprising:

a first component for presenting a string including the ciphertext string of a message for decryption;

a second component for partitioning said ciphertext string into a plurality of ciphertext segments;

a third component for concurrently presenting said plurality of ciphertext segments to a plurality of decryption modes;

a fourth component for obtaining a different secret random number per ciphertext segment in the same manner as at the system for segmented encryption;

a fifth component for decrypting each ciphertext segment using said different secret random number per ciphertext segment to obtain a plaintext segment, using a system for parallel decryption that is the inverse of a system for parallel encryption that provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, wherein said single cryptographic primitive is an ℓ-bit block cipher using a secret key, and using a non-cryptographic Manipulation Detection Code function for verifying integrity of the plaintext blocks of each plaintext segment; and

a sixth component for verifying the integrity of each plaintext segment and for each plaintext segment, outputting either the plaintext segment if the integrity verification passes, or an error indicator.

108. A program product for a parallel encryption for providing both data confidentiality and integrity for a message, that updates a ciphertext string incrementally, including machine-readable code for performing the following method steps:

receiving an input plaintext string comprising a message;

generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks;

performing a plaintext randomization function over said plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length;

processing each of said hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks;

performing a hidden ciphertext randomization function over said plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length; and

further including machine-readable code for perfoming the following method steps:

receiving an input plaintext string;

generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

receiving an input ciphertext string including a plurality of $n+1$ equal-size blocks of the ciphertext of ℓ bits in length, wherein the $n+1$ block of the ciphertext corresponds to an MDC block for said plaintext string;

receiving a new ℓ-bit plaintext block to replace an ℓ-bit plaintext block at index i;

creating a new MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks and the new ℓ-bit plaintext block;

performing the same plaintext randomization function as that used at a parallel encryption method over said new ℓ-bit plaintext block and the new MDC block to create two new hidden plaintext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index $n+1$ for the new MDC block;

processing each of said two new hidden plaintext blocks by a block cipher using said secret key (K) to obtain two new hidden ciphertext blocks;

performing the same hidden ciphertext randomization function as that used at a parallel encryption method over said two new hidden ciphertext blocks to create two new output ciphertext blocks each

of ℓ bits in length using index i for the new ℓ-bit plaintext block and index $n+1$ for the new MDC block;

replacing in the input ciphertext string, the input ciphertext block at index i with the output ciphertext block for the new ℓ-bit plaintext block and replace the input ciphertext block at index $n+1$ with the output ciphertext block for the new MDC block, to create a new ciphertext string; and

outputting the new ciphertext string.

109. The program product as defined in claim 108, wherein the program code for causing the performance of the step of generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises code for:

padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and

partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

110. The program product of claim 108 including machine-readable code for performing the method steps:

receiving a plurality of new ℓ-bit plaintext blocks to replace a plurality of ℓ-bit plaintext blocks at said plaintext string at index i; and

providing a parallel encryption method that outputs a ciphertext string incrementally for each of the said plurality of new ℓ-bit plaintext blocks.

111. A program product for parallel encryption method for providing both data confidentiality and integrity for a message, including machine-readable program code for causing a machine to perform the method steps:

receiving an input plaintext string comprising a message;

generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length;

creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks;

performing a plaintext randomization function over said plurality of equal-sized blocks of the plaintext and the MDC block using a different plaintext index for each equal-sized block and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length;

processing each of said hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks;

performing a hidden ciphertext randomization function over said plurality of hidden ciphertext blocks using a different ciphertext index for each hidden ciphertext block to create a plurality of output ciphertext blocks each of ℓ bits in length; and

further including machine-readable program code for performing an out-of-order decryption method for the parallel encryption method, which provides both data confidentiality and integrity, including code for:

receiving a string including a plurality of $n+1$ ℓ-bit ciphertext blocks for decryption;

selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of

the selected n + 1 ciphertext blocks using said ciphertext index to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length;

processing each of said hidden ciphertext blocks with the inverse of the block cipher used at an encryption method using said secret key (K) to obtain a plurality of hidden plaintext blocks; and

performing an inverse plaintext randomization function over said plurality of hidden plaintext blocks using said plaintext index to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ-bit length;

creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks in the same manner as at a parallel encryption method;

verifying integrity of the plaintext blocks by comparing said created MDC decryption block with the decrypted MDC block;

outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and

outputting a failure indicator if the integrity verification fails.

112. The program product as defined in claim 111, wherein the program code for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string comprises code for:

padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and

partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

113. A system for a parallel encryption for providing both data confidentiality and integrity for a message, that updates a ciphertext string incrementally, comprising:

a first component for receiving an input plaintext string comprising a message;

a second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

a third component for creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks;

a fourth component for performing a plaintext randomization function over said plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length;

a fifth component for processing each of said hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks;

a sixth component for performing a hidden ciphertext randomization function over said plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length; and

further comprising:

a seventh component for receiving an input plaintext string;

an eight component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

a ninth component for receiving an input ciphertext string including a plurality of $n+1$ equal-size blocks of the ciphertext of ℓ bits in length, wherein the $n+1$ block of the ciphertext corresponds to an MDC block for said plaintext string;

a tenth component for receiving a new ℓ-bit plaintext block to replace an ℓ-bit plaintext block at index i;

an eleventh component for creating a new MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks and the new ℓ-bit plaintext block;

a twelfth component for performing the same plaintext randomization function as that used at a parallel encryption method over said new ℓ-bit plaintext block and the new MDC block to create two new hidden plaintext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index $n+1$ for the new MDC block;

a thirteenth component for processing each of said two new hidden plaintext blocks by a block cipher using said secret key (K) to obtain two new hidden ciphertext blocks;

a fourteenth component for performing the same hidden ciphertext randomization function as that used at a parallel encryption method over said two new hidden ciphertext blocks to create two new output ciphertext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index $n+1$ for the new MDC block;

a fifteenth component for replacing in the input ciphertext string, the input ciphertext block at index i with the output ciphertext block for the new ℓ-bit plaintext block and replace the input ciphertext block at index $n+1$ with the output ciphertext block for the new MDC block, to create a new ciphertext string; and

a sixteenth component for outputting the new ciphertext string.

114. The system as defined in claim 113, wherein said second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises:

a component for padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and

a component for partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

115. The system of claim 113, further comprising:

a component for receiving a plurality of new ℓ-bit plaintext blocks to replace a plurality of ℓ-bit plaintext blocks at said plaintext string at index i; and

a component for providing a parallel encryption method that outputs a ciphertext string incrementally for each of the said plurality of new ℓ-bit plaintext blocks.

116. A system for parallel encryption method for providing both data confidentiality and integrity for a message, comprising:

a first component for receiving an input plaintext string comprising a message;

a second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string;

a third component for partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length;

a fourth component for creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of said equal-size blocks;

a fifth component for performing a plaintext randomization function over said plurality of equal-sized blocks of the plaintext and the MDC block using a different plaintext index for each equal-sized block and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length;

a sixth component for processing each of said hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks;

a seventh component for performing a hidden ciphertext randomization function over said plurality of hidden ciphertext blocks using a different ciphertext index for each hidden ciphertext block to create a plurality of output ciphertext blocks each of ℓ bits in length; and

further comprising for performing an out-of-order decryption method for the parallel encryption method, which provides both data confidentiality and integrity:

an eighth component for receiving a string including a plurality of $n+1$ ℓ-bit ciphertext blocks for decryption;

a ninth component for selecting $n+1$ ciphertext blocks from said plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks using said ciphertext index to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length;

a tenth component for processing each of said hidden ciphertext blocks with the inverse of the block cipher used at an encryption method using said secret key (K) to obtain a plurality of hidden plaintext blocks; and

an eleventh component for performing an inverse plaintext randomization function over said plurality of hidden plaintext blocks using said plaintext index to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ-bit length;

a twelfth component for creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks in the same manner as at a parallel encryption method;

a thirteenth component for verifying integrity of the plaintext blocks by comparing said created MDC decryption block with the decrypted MDC block;

a fourteenth component for outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and

a fifteenth component for outputting a failure indicator if the integrity verification fails.

117. The system as defined in claim 116, wherein said second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string comprises:

a component for padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and

a component for partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.